International Journal of Computer Science Engineering
and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 4, Issue 5, Oct 2014, 13-22

IMPLEMENTATION OF A CONVERGENT B-CVA BASED ML DECODER USING CONVOLUTIONAL TAIL BITING TRELLIS

MOMINA KOUSER[1] & CHANDRAKALA. V[2]

[1]PG Student, Department of Telecommunication, Dr. AIT, Bangalore, Karnataka, India

[2]Assistant Professor, Department of Telecommunication, Dr. AIT, Bangalore, Karnataka, India

ABSTRACT

Tail-biting trellises are the simplest of decoding graphs with cycle. Basically, trellis representations not only reveal the code structure, but also lead to efficient trellis based decoding algorithms. Existing Circular Viterbi Algorithms are non-convergent and sub-optimal. In this paper, decoding of convolution code is done over a Rayleigh channel and it is shown that the net path metric of each tail-biting path is lower-bounded during the decoding process of the CVA. This lower bounding property can be applied to remove unnecessary iterations of the existing CVA and results in a bounded or convergent CVA based Maximum Likelihood (ML) decoder. Comparison between the existing two phase ML decoder and the convergent CVA decoder is done over a Rayleigh channel to show that the proposed algorithm has higher efficiency and lower decoding complexity.

KEYWORDS: CVA, Maximum Likelihood (ML) Decoder

INTRODUCTION

Tail-Biting Convolutional Codes (TBCCs) are simple and powerful forward error correction codes. There are many advantages of using TBCC over the conventional Zero-Tail Convolutional Codes (ZTCC) and some block codes. Convolutional Tail-Biting Codes (CTBC) can overcome the loss on the code rate, and induces less performance degradation [7, 5]. In the trellis of CTBC, there is a one-to-one correspondence between a codeword and a path with the same initial and final state, which is called a tail-biting path. If the number of initial states (equivalently, final states) of the convolutional tail-biting code is N, the trellis is composed of N subtrellises with the same initial and final state. These subtrellises are called tail-biting subtrellises, or simply subtrellises, and will be denoted by T_s for the ith subtrellis. Since Maximum Likelihood (ML) decoding for TBCCs involves high complexity calculations, researchers have proposed several sub-optimal decoding algorithms for practical applications. [2, 12]. Among them, the Wrap-Around Viterbi Algorithm (WAVA) is the one which iteratively applies the Viterbi algorithm (VA) in a wrap-around style because the TBCC is circular. At the end of each iteration, the algorithm checks some sufficient condition for an optimal solution to determine if iteration is necessary.

Recently, an ML decoding algorithm of practical decoding complexity was proposed. This scheme has two phases. In the first phase, the Viterbi algorithm is applied to the trellis of the convolutional tail-biting code to obtain the trellis information. Based upon the trellis information, the algorithm A* is then performed on all subtrellises in the second phase to yield the ML decision. It has been shown that the decoding complexity can be reduced from N Viterbi Algorithm trials to equivalently 1.3 VA trials without sacrificing the optimality in performance. This approach achieves near-optimal performance with an acceptable complexity. Two iterations are usually enough for most TBCCs. The circular characteristic

of the TBCC suggests that it is unnecessary to start the decoding process at the beginning of a received codeword.

The circular Viterbi algorithm (CVA)-based decoder greatly reduces the implementation complexity of a decoder for tail-biting trellises and provides near-optimal block error rate performance. However, the decoding process of the CVA is non-convergent and sub-optimal [4, 7]. In this paper, a CVA-based ML decoder for tail-biting trellises is introduced.

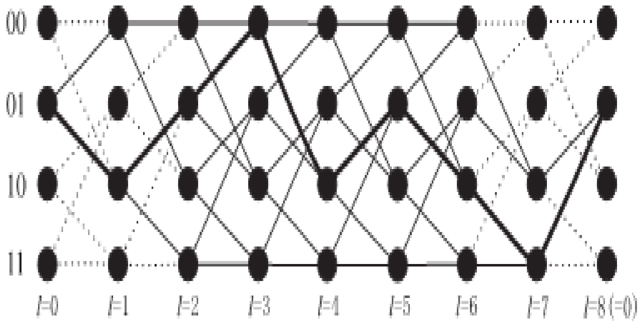

Figure 1: The Tail-Biting Trellis of the (8, 4) Convolutional Code with Octal Generator Polynomials of {7, 5}

In the bounded CVA algorithm the decoding is done using Rayleigh channel. Here during the decoding, the lower bound of the net path metric of each tail-biting path can be obtained to exclude impossible starting state candidates resulting in convergence of the existing CVA. In addition, the net path metric of survivor paths can be used to terminate redundant searches without performing a full Viterbi iteration, hence resulting in less number of iterations.

PROPOSED ALGORITHM

Algorithm Explanation

Consider an example of decoding over a Rayleigh channel with tail-biting convolutional codes of rate b/c, the length of information bits is bL and the length of the corresponding codeword is cL. Binary code bits $v_l^{(j)}$are mapped to $x_l^{(j)} = (1 - 2v_l^{(j)})\sqrt{E_s}$ with binary phase-shift keying (BPSK) modulation, where $0 \leq j \leq c - 1$.and $0 \leq l \leq L - 1$. Without loss of generality, signal energy Es is normalized to 1. After passing through an reyleigh channel, the corresponding received symbols is r_l^j. The log-likelihood ratio of r_l^j is given by $\wedge\left(r_l^{(j)}\right) = 4r_l^{(j)}$ $N0$, where for $l \geq L, x_l^{(j)} = x_{(l)_L}^{(j)}, \wedge\left(r_l^{(j)}\right) = \wedge\left(r_{(l)_L}^{(j)}\right)$, and $(l)_L = l$ mod. During the decoding process of the CVA, the accumulated path metric of the survivor path entering state *s* at location *l* in the *i*th iteration is

$$M_l^i(s) = \sum_{k=0}^{(i-1)L+1} \sum_{j=0}^{c-1} \left(x_k^{(j)} \wedge \left(r_k^{(j)}\right)\right), i \geq 1 \tag{1}$$

The weighted Hamming distance between $\wedge\left(r_l^{(j)}\right)$ and $x_l^{(j)}$ can be defined as in [8]:

$$D\left(\wedge\left(r_l^{(j)}\right), x_l^{(j)}\right) = \begin{cases} \left| \wedge\left(r_l^{(j)}\right) \right|, if\ sgn(\wedge\left(r_l^{(j)}\right)) \neq x_l^{(j)} \\ 0, \text{otherwise}, \end{cases} \tag{2}$$

Where, $(\wedge\left(\boldsymbol{r}_l^{(j)}\right))$ denotes the sign of $\wedge\left(\boldsymbol{r}_l^{(j)}\right)$,Based on (1) and (2), the ML decoding on the tail-biting trellis is equivalent to solving the following equation:

$$\begin{aligned}\hat{x} &= \arg\max_{\mathbf{x}} \sum_{k=(i-1)L}^{iL-1} \sum_{j=0}^{c-1} \left(x_k^j \cdot \Lambda\left(r_k^{(j)}\right)\right) \\ &= \arg\max_{\mathbf{x}} \sum_{k=(i-1)L}^{iL-1} \sum_{j=0}^{c-1} \left|\Lambda\left(r_k^{(j)}\right)\right| - 2D\left(\Lambda\left(r_k^{(j)}\right), x_k^{(j)}\right) \\ &= \arg\min_{\mathbf{x}} \sum_{k=(i-1)L}^{iL-1} \sum_{j=0}^{c-1} D\left(\Lambda\left(r_k^{(j)}\right), x_k^{(j)}\right).\end{aligned} \tag{3}$$

The term, $\left|\wedge\left(\boldsymbol{r}_l^{(j)}\right)\right|$ can be ignored in the third line of (3) since it is independent of specific codewords x and consequently is a constant for all paths on the tail-biting trellis. Denote by $\boldsymbol{P^i}(\boldsymbol{\beta^i}(\boldsymbol{s}), \boldsymbol{s})$ the survivor path that connects state $\boldsymbol{\beta}_i(s)$ of $\boldsymbol{S_0}$ with state *s* of $\boldsymbol{S_l}$ in the *i*th iteration. The corresponding net path metric of $\boldsymbol{P^i}(\boldsymbol{\beta^i}(\boldsymbol{s}), \boldsymbol{s})$ can be derived from (1) and (3):

$$\boldsymbol{M^i_{net}}(\boldsymbol{\beta^i}(\boldsymbol{s}), \boldsymbol{s}) = \boldsymbol{M^i_l}(\boldsymbol{s}) - \boldsymbol{M^i_0}\left(\boldsymbol{\beta^i}(\boldsymbol{s})\right) \tag{4}$$

Since the initial path metrics are different $\boldsymbol{M_0^{i+1}}(\boldsymbol{s'})$from for each state $\boldsymbol{s'} \in \boldsymbol{S_0}$, different survivor paths can be obtained in each iteration. Denote by $\boldsymbol{P_i}$ the ML path obtained in the *i*th iteration, where the ML path obtained from the first iteration has the least net path metric among all possible survivor paths [4].Similarly, the ML tail-biting path obtained in the *i*th iteration is denoted by $\widetilde{\boldsymbol{P}}^{\boldsymbol{i}}$, which has the net path metric $\widetilde{\boldsymbol{M}}^{\boldsymbol{i}}$. Among the set of tail-biting paths $\widetilde{\boldsymbol{M}}^{\boldsymbol{i}\,\prime}$ *(s, s))* | ∀*s* ∈ $\boldsymbol{S_0}$, *i* ≥ 1}, the optimal tail-biting path and its net path metric are denoted by $\widetilde{\boldsymbol{P}}^{\boldsymbol{o}}$ and $\widetilde{\boldsymbol{M}}^{\boldsymbol{o}}$, respectively.

Lower Bounds of the Net Path Metrics

The existing CVA-based decoder is non-convergent, and it cannot guarantee that the tail-biting path obtained is optimal when the decoding process is terminated. In order to design a convergent CVA based ML decoder on tail-biting trellises over a Rayleigh channel, further information needs to be obtained from the decoding process of CVA. Based on the characteristics of CVA, a lower bound of the net path metric of each tail-biting path is derived. This observation is summarized as: Let $\tilde{P}$(s, s) denote the ML tail-biting path on the sub-tail-biting trellis of state s, and the corresponding net path metric i $\widetilde{M}$(s, s) where s ∈ s_0. Define B(s) as

$$B(s)=\max_{i\geq 1} \boldsymbol{M^i_L}(\boldsymbol{s}) - \boldsymbol{M^i_0}(\boldsymbol{s}), \tag{5}$$

then B(s) ≤ $\widetilde{M}$(s, s), i.e., B(s) is the lower bound of the metrics of all paths on the sub-tail-biting trellis of state s.

Proof. The tail-biting trellis defined on a circular time axis can be split at section $l = 0$ and duplicated on the time axis head-tail. Conventional CVA then becomes a general Viterbi decoder composed of several length *L* decoding sections, where the Viterbi algorithm searches on the duplicated trellis by recording and repeating the received symbols. Consequently, combining (1) and (3), we find that the survivor path *Pi(βi(s), s)* has the minimum accumulated path metric among all possible paths $P_i(s*, s)$, s*, $\boldsymbol{\beta^l}(\boldsymbol{s}) \in \boldsymbol{S_0}$, s ∈ $\boldsymbol{S_l}$ and $0 \leq l \leq L-1$

Consequently,

$$M_l^i(s) = M_0^i(\beta^i(s) + M_{\text{net}}^i(\beta^i(s), s)$$

$$\leq M_0^i(s^*) + M_{\text{net}}^i(s^*, s) \tag{6}$$

Since (6) holds for $0 \leq l \leq L-1$, we know that for any $s \in S_L$, (6) also holds. Then for the ML tail-biting path,

$\tilde{P}(s, s)$ on the sub-tail-biting trellis of state s, from (6), is given that

$$M_L^i(s) - M_L^i(s) \leq M_{\text{net}}^i(s,s) \leq \widetilde{M}(s, s),\ i \geq 1 \tag{7}$$

Since, B(s) = $\max_{i\geq 1} \boldsymbol{M_L^i(s)} - \boldsymbol{M_0^i(s)}$ then combining (7), we have

$$B(s) \leq \widetilde{M}(s, s) \tag{8}$$

Since, $\tilde{P}(s, s)$ is the ML tail-biting path over Rayleigh channel on the sub-tailbiting trellis of state *s*, from (3) and (8), the conclusion is that *B(s)* is a lower bound of the metrics of all paths on the sub-tail-biting trellis of state *s*. The lower bound *B(s)* is updated as iterations continues and a more precise estimation of *B(s)* can be obtained if more iterations are performed on the tail-biting trellis. According to (8), the maximal value of *B(s)* is $\widetilde{M}(s, s)$. Thus the decoding complexity of CVA on the tail-biting trellis can be reduced by removing redundant computations and iterations during the decoding process (using Rayleigh) and control the convergence of CVA based ML decoder. The improvements of the proposed decoder can be summarized into the following two aspects. Firstly, during the decoding if the net path metric $M_{\text{net}}^i(\beta^i(s), s)$ of survivor path $P_i(\beta^i(s), s)$ is not less and $\widetilde{M^0}$ is the optimal tail-biting path obtained in the first $i-1$ iterations and $s \in S_l$, all searches that follow state *s* can be terminated (refer to Figure 2). In this case, the net path metric of any survivor path that starts from state $\beta^i(s)$ and passes through state *s* is not less than, where $\widetilde{M^0}$ is the optimal tail-biting path obtained in the first $i-1$ iterations and $s \in S_l$, all searches that follow state *s* can be terminated (refer to Figure 2). In this case, the net path metric of any survivor path that starts from state $\beta^i(s)$ and passes through state *s* is not less than $\widetilde{M^0}$. Secondly, denote by S_C^i the set of survivor starting state candidates in the *i*th iteration of the CVA, i.e.

$$\boldsymbol{S_C^i} = \left\{ \boldsymbol{s} \,\middle|\, \boldsymbol{B(s)} < \widetilde{\boldsymbol{M^0}} \right\} \tag{9}$$

Here $\boldsymbol{S_C^1} = \boldsymbol{S_0}$. For $\forall\, \boldsymbol{s} \in \boldsymbol{S_C^i}$, **if** $\boldsymbol{B(s)} \geq \widetilde{\boldsymbol{M^0}}$, then from Lemma 1, it is given that $\widetilde{\boldsymbol{M}}(\boldsymbol{s},\boldsymbol{s}) \geq \widetilde{\boldsymbol{M^0}}$.

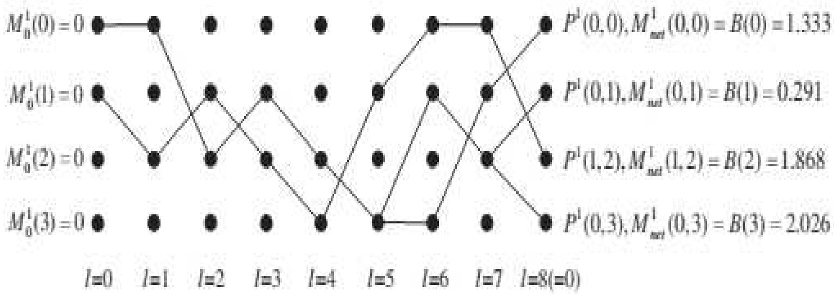

Figure 2(a): Decoding Process of the B-CVA ML Decoder

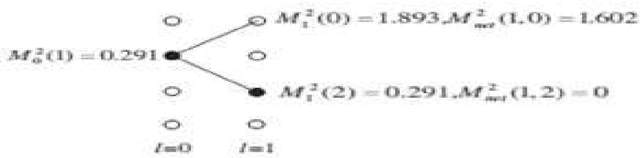

Figure 2(b): Decoding Process 1

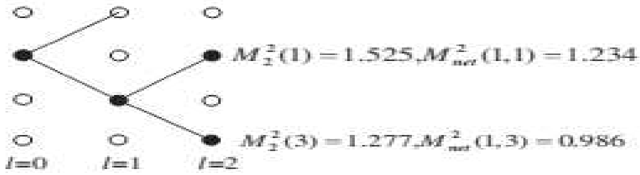

Figure 2(c): Decoding Process 2

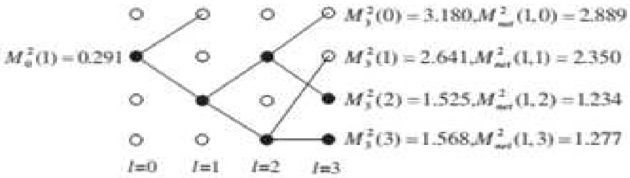

Figure 2(d): Decoding Process 3

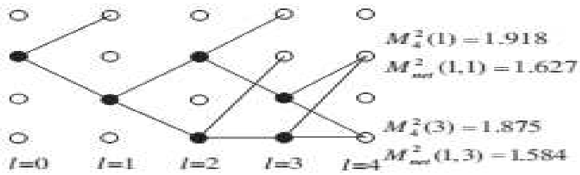

Figure 2(e): Decoding Process 4

As the decoding iterations continue, the search space on the tail-biting trellis over a rayleigh channel shrinks and the decoder will converge to the global optimal solution eventually.

CVA Based ML Decoder on Tail-Biting Trellis

The above decoding process on tailbiting trellis over a rayleigh channel is summarized as follows.

- Firstly, state s with bound B(s) > $\widetilde{M}^{O}$ is deleted from s_c^i in step 2.3 since the ML tail-biting path on the sub-tail biting trellis of state is not better than $\tilde{P}^{O}$.
- Secondly, after the *(i* + 1*)*th iteration, if no state has been deleted from s_c^{i+1} i.e $\forall_s \in s_c^{i+1}, B(s) < \widetilde{M}^{O}$, then equation $s_c^{i+1} = s_c^{i+2}$ holds and the Viterbi algorithm will be performed on the sub-tail-biting trellis of state s† in step 2.4, where $\boldsymbol{s}\dagger \in s_c^{i+1}$.If $\widetilde{M}\ \ (s\dagger, s\dagger) \geq \widetilde{M}^{O}$.this indicate that $\widetilde{P}(s\dagger, s\dagger)$ is not better than $\widetilde{P_o}$. Then state s† can be deleted from s_c^{i+1} since the ML tail biting path on its sub tail biting trellis has been found.
- Thirdly, When s_c^i is empty, the decoder converges to the global optimal solution that was recorded.

Algorithm Steps

Table 1

1	Initialization $\boldsymbol{s_c^1} \leftarrow \boldsymbol{s_0}, \boldsymbol{M_0^1} \leftarrow \boldsymbol{0}$ for all $\boldsymbol{s} \in \boldsymbol{s_c^1}$, $\boldsymbol{\widetilde{M}^O} \leftarrow +\infty$;
2	For iteration $\boldsymbol{i}, \boldsymbol{i} \geq \boldsymbol{1}$:
2.1	Perform Viterbi algorithm on the tail-biting trellis with the set of starting state candidates s_c^i, during the decoding process,$\forall P^i(\beta^i(s'), s')$, where $s' \in s_l$.if $M_{net}^i(\beta^i(s'), s' \geq \widetilde{M}^O$,terminate the search that follows state s' as mentioned above;
2.2	Find ML tail-biting path P^iif it exists, update $\widetilde{P^0}, \widetilde{M^0}$, with $(\widetilde{P^\iota}, \widetilde{M^\iota})$ if $\widetilde{M^\iota} < \widetilde{M^0}$;
2.3	For $\forall s \in s_c^i$, calculate *Bi(s)* = $M_L^i(s) - M_O^i(s)$ and update *B(s)* with *Bi(s)* if *Bi(s)* > *B(s)*; generate s_c^{i+1}, by excluding state s with *B(s)* $\geq M^0$ from set s_c^i,

Table 1: Contd.,

2.4	If $s_c^{i+1} = \emptyset$,go to step 3;else if $s_c^{i+1} = s_c^i$,find the starting state $s\dagger$ of ML path $\overline{P^i}$ and perform viterbi algorithm on the sub-tail biting trellis of state $s\dagger$, step 2.1 is also applied in this case; delete $s\dagger$ from s_c^{i+1};Update $\widetilde{P^0}$ with $\widetilde{P^i}, (s\dagger, s\dagger)$ if $\widetilde{M}(s\dagger, s\dagger) < M^0$;
2.5	$\forall s \in s_c^{i+1}$:$M_0^{i+1}(s) \leftarrow M_L^i(s); i \leftarrow i+1$;go back to step 2;
3	Output the codeword associated with $\widetilde{P^0}$;

SIMULATION RESULTS

The existing ML decoder is 2 phase ML decoder and the proposed is B-CVA ML decoder. Here ES is the existing system and PS is the proposed system. In the present paper of "low complexity CVA based ML decoder" X Wang has shown that the decoding efficiency can be improved by lower bounding the path metric on the trellis using AWGN channel. It can be observed from the below graphs how efficient and less complex does a bounded CVA based ML decoder is, when compared to existing 2 phase ML decoder over a Rayleigh channel. From the below table 2 and the graphs it is clear that the proposed decoder (B-CVA based ML decoder) gives the higher efficiency compared to existing WAVA (Wrap Around Viterbi Algorithm) and the two phase ML decoder.

Table 2: BER Comparison of Proposed System and the Existing System over a Rayleigh Channel

BLER Performance	0.0dB	1.0dB	2.0dB	3.0dB	4.0dB
WAVA	3.95×10^{-1}	1.49×10^{-1}	2.80×10^{-2}	2.70×10^{-3}	1.06×10^{-4}
B-CVA based ML decoder	2.95×10^{-1}	1.48×10^{-2}	2.66×10^{-2}	2.50×10^{-3}	0.96×10^{-4}

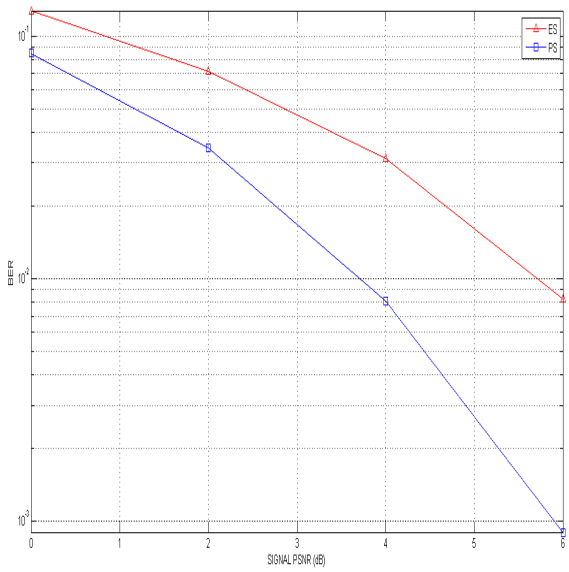

Figure 3: SNR v/s BER Plot for (64, 32) Code

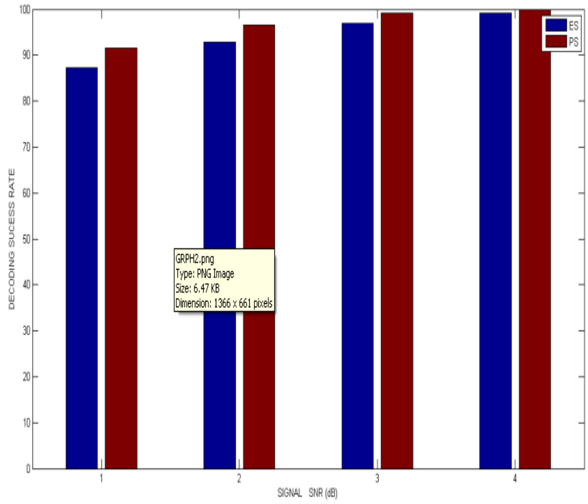

Figure 4: SNR v/s Success Rate Plot for (64, 32) Code

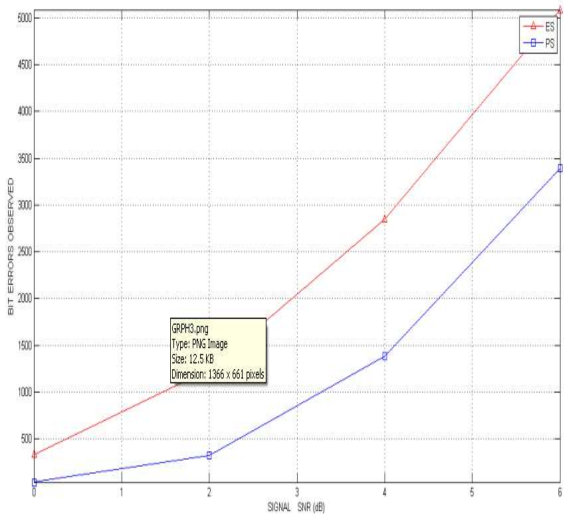

Figure 5: SNR v/s Bit Errors Observed Plot for (64, 32) Code

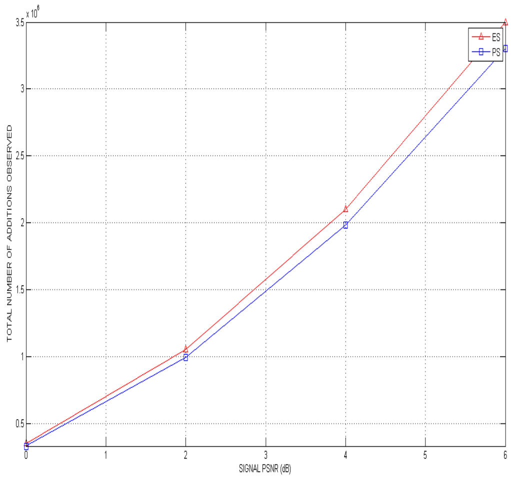

Figure 6: Total Number of Additions Required for (64, 32) Code

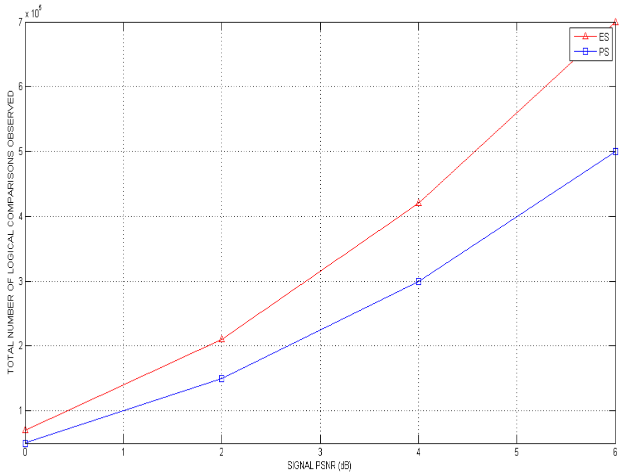

Figure 7: Total Number of Logical Comparisons Required for (64, 32) Code

CONCLUSIONS

In this paper, a convergent CVA-based ML decoder for tailbiting trellises over Rayleigh fading channel is proposed. Here the lower bounding property of the tail-biting path is used to control the decoding process of existing CVA. Simulation results show that, on tail-biting trellises, the B-CVA based ML decoder exhibits high decoding efficiency and lesser decoding complexity during the decoding process when compared to existing ML decoders.

ACKNOWLEDGEMENTS

I would like to thank my project guide and mentor Chandrakala. V, Assistant Professor, Dept. of Telecommunication, Dr. Ambedkar Institute of Technology, Bangalore, for her guidance, suggestions and support throughout every phase of this project.

REFERENCES

1. A Calderbank, et al. (1999). Minimal tailbiting trellises: Golay code and more: *IEEE Transaction on Information Theory*, 45(5), 1435-1455
2. P Stahl, et al. (1999). Optimal and near-optimal encoders for short and moderate-length tailbiting trellises: *IEEE Transaction on Information Theory*, 45(7), 2562–2571
3. H Gluesing-Luerssen, E Weaver. (2011). Linear tail-biting trellises: characteristic generators and the BCJR-construction: *IEEE Transaction on Information Theory*, 57(2), 738–751
4. X Wang, et al. (2011). An efficient CVA-based decoding algorithm for tail-biting Codes: *IEEE Global Telecommunications Conference*, 14, 1–5
5. P Shankar, et al. (2007) Efficient convergent maximum likelihood decoding on tail-biting: *IEEE Transaction on Information Theory*, 47, 1–15
6. H Pai, Y Han, et al. (2008). Low-complexity ML decoding for convolutional tail-biting codes: *IEEE Transaction on Information Theory*, 12(12), 883–885

7. Shao, et al. (2003). Two decoding algorithms for tailbiting codes: *IEEE Transaction on Information Theory*, 51(10), 1658–1665

8. IE Bocharova, et al. (2004). BEAST decoding for block codes: *European Transaction on Information Theory*, 15, 297–305

9. R. V. Cox and C. E. W. Sundberg. (2011). An efficient adaptive circular viterbi algorithm for decoding generalized tailbiting convolutional codes: *IEEE Transaction on Information Theory*, 51(10), 1658–1665

10. Jorge Ortin, et al. (2011). Simplified Circular Viterbi Algorithm for Tailbiting Convolutional Codes: *Aragon Institute for Engineering Research (I3A),* 57(2), 738–751

11. R. V. Cox and C. E. W. Sundberg.(2013). A simplified circular viterbi algorithm for decoding generalized tailbiting convolutional codes: *IEEE Transaction on Vehicular Technology*, 43, 57–68

12. X Wang, et al. (2013) An Efficient ML Decoder For Tail-Biting Codes Based On Circular Trap Detection: *IEEE Transaction on communication*, 61, 1-4